Redistricting Homework 5

Mark Schulist

November 20, 2024

[1]:
```
from gerrychain import Graph
from gerrychain.tree import recursive_tree_part
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
import geopandas as gpd
```

[2]:
```
az_bg_connected = Graph.from_json("data/az-bg-connected.json")
```

[3]:
```
POPULATION_KEY = "TOTPOP20"
```

[4]:
```
total_population = sum(
    az_bg_connected.nodes[node][POPULATION_KEY] for node in az_bg_connected.
 ↪nodes
)
```

0.1 Districting Plan

[5]:
```
n_districts = 30
ideal_population = total_population / n_districts

initial_plan = recursive_tree_part(
    az_bg_connected, range(n_districts), ideal_population, POPULATION_KEY, 0.
 ↪02, 10
)
```

```
/Users/mschulist/miniconda3/envs/cse217a/lib/python3.10/site-
packages/gerrychain/tree.py:704: BipartitionWarning:
Failed to find a balanced cut after 1000 attempts.
If possible, consider enabling pair reselection within your
MarkovChain proposal method to allow the algorithm to select
a different pair of districts for recombination.
  warnings.warn(
```

0.2 Cut Edges

[6]:
```
cutedges = 0
for e in az_bg_connected.edges():
    if initial_plan[e[0]] != initial_plan[e[1]]:
        cutedges += 1
print("Number of cutedges in initial_plan: ", cutedges)
```

```
Number of cutedges in initial_plan:  1635
```

0.3 Latino Majority Districts

[7]:
```
HISPANIC_POPULATION_KEY = "HISP20"
```

[8]:
```
district_populations = np.zeros(n_districts)

for node in az_bg_connected.nodes:
    district_populations[initial_plan[node]] += az_bg_connected.nodes[node][
        POPULATION_KEY
    ]

district_populations
```

[8]:
```
array([239733., 234810., 239597., 238292., 237506., 237557., 241176.,
       242802., 238198., 235464., 239566., 238249., 240228., 234415.,
       240938., 235486., 236381., 239185., 237794., 237962., 242249.,
       234305., 236331., 239021., 238217., 242124., 239679., 241526.,
       235181., 237530.])
```

[9]:
```
latino_district_populations = np.zeros(n_districts)

for node in az_bg_connected.nodes:
    latino_district_populations[initial_plan[node]] += az_bg_connected.
 ↪nodes[node][
        HISPANIC_POPULATION_KEY
    ]

latino_district_populations
```

[9]:
```
array([185047.,  65783.,  70394.,  41283.,  45447., 106287.,  41956.,
        43197.,  51110.,  33866.,  97820.,  76229., 110676.,  43985.,
       103681.,  86999., 151798.,  52912., 104447.,  69125.,  58655.,
       121776.,  30573.,  67150.,  16385.,  43716., 138990.,  51727.,
        48093.,  33146.])
```

[10]:
```
LATINO_POPULATION_proportion = latino_district_populations /␣
 ↪district_populations
```

```
num_majority_latino_districts = sum(LATINO_POPULATION_proportion > 0.5)

num_majority_latino_districts
```

[10]: 4

0.4 Pretty Map

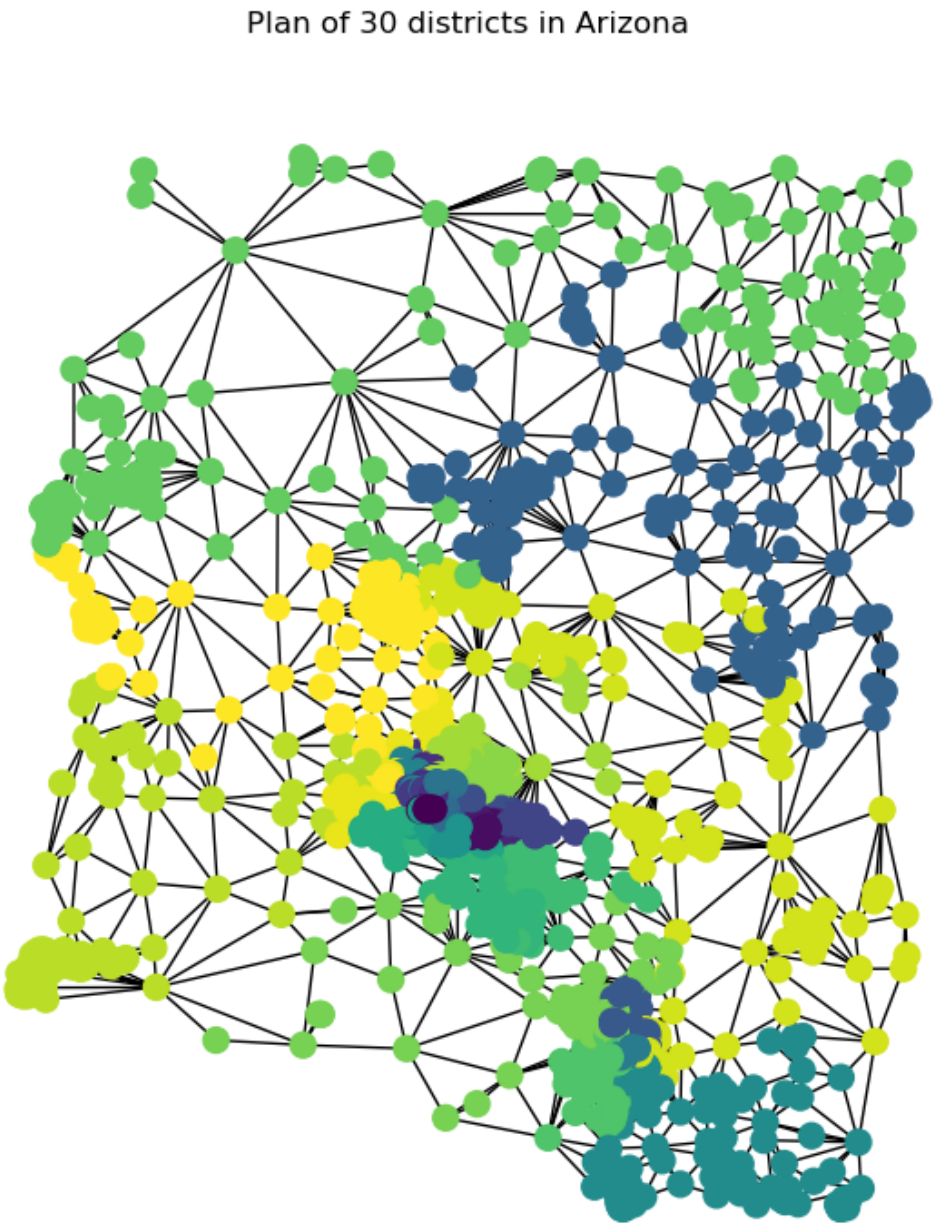
[11]:
```
LONGITUDE_KEY = "INTPTLON20"
LATITUDE_KEY = "INTPTLAT20"
```

[12]:
```
position_dict = {
    v: (
        float(az_bg_connected.nodes[v][LONGITUDE_KEY]),
        float(az_bg_connected.nodes[v][LATITUDE_KEY]),
    )
    for v in az_bg_connected.nodes
}
```

[13]:
```
x = np.array([initial_plan[v] for v in az_bg_connected.nodes()])
```

[14]:
```
plt.figure(figsize=(6, 7))
node_colors = np.array([initial_plan[v] for v in az_bg_connected.nodes()])
nx.draw(
    az_bg_connected,
    node_size=100,
    node_color=node_colors,
    pos=position_dict,
)
plt.title(f"Plan of {n_districts} districts in Arizona")
plt.show()
```

```
/Users/mschulist/miniconda3/envs/cse217a/lib/python3.10/site-
packages/IPython/core/pylabtools.py:77: DeprecationWarning: backend2gui is
deprecated since IPython 8.24, backends are managed in matplotlib and can be
externally registered.
  warnings.warn(
```



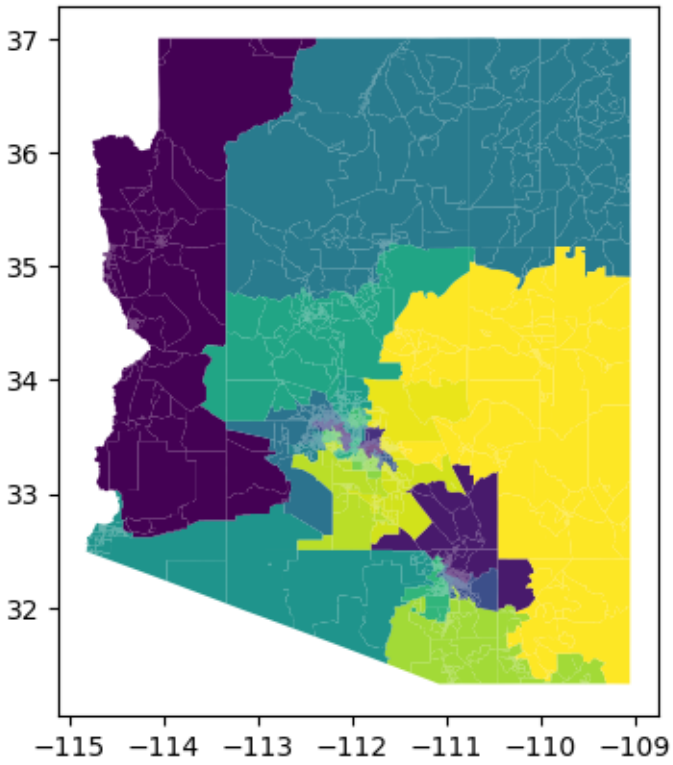
0.5 Cool Map

[15]:
```
ip_list = [[i, p] for (i, p) in initial_plan.items()]

plan_gdf = gpd.GeoDataFrame(ip_list, columns=["graph_id", "district"])
```

[16]:
```
geoids = []
for i, row in plan_gdf.iterrows():
    geoids.append(int(az_bg_connected.nodes[row["graph_id"]]["GEOID20"]))

plan_gdf["GEOID"] = geoids
```

[17]:
```
az_bg_map: gpd.GeoDataFrame = gpd.read_file("data/tl_2023_04_bg")

az_bg_map["GEOID"] = az_bg_map["GEOID"].astype(int)

az_bg_map = az_bg_map.merge(plan_gdf, on="GEOID")
```

[85]:
```
az_bg_map.plot(column="district")
plt.show()
```



0.6 ReComb (not for HW)

[22]:
```
from gerrychain import Partition
from gerrychain.updaters import cut_edges, Tally
from gerrychain.proposals import recom
```

[23]:
```
inital_partition = Partition(
    graph=az_bg_connected,
    assignment=initial_plan,
    updaters={
        "our_cut_edges": cut_edges,
        "population": Tally(POPULATION_KEY, alias="population"),
        "district_latino_population": Tally(
            HISPANIC_POPULATION_KEY, alias="district_latino_population"
        ),
    },
)
```

[ ]:
```

```